AD-A062 495 GEORGIA INST OF TECH ATLANTA SCHOOL OF INFORMATION A--ETC F/G 12/1
SPACE-TIME TRADEOFFS IN STRUCTURED PROGRAMMING: AN IMPROVED COM--ETC(U)
AUG 78 R A DEMILLO, S C EISENSTAT, R J LIPTON DAHC04-74-G-0179

UNCLASSIFIED GIT-ICS-78/03 ARO-14690.6-EL NL

1 OF 1

AD
A062495

END
DATE
FILMED
3 -- 79
DDC

MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

(9) Technical Rept.,

(14) GIT-ICS-78/03

(6)

SPACE-TIME TRADEOFFS IN STRUCTURED PROGRAMMING: AN IMPROVED COMBINATORIAL EMBEDDING THEOREM,

(10) RICHARD A. DEMILLO,*
STANLEY C. EISENSTAT**
RICHARD J. LIPTON**

(15) DAHC04-74-G-0179,
DAAG29-76-G-0338

(11) Aug[illegible]78

(12) 14p.

(18) ARO

(19) 14690.6-EL

* School of Information and Computer Science
Georgia Institute of Technology
Atlanta, Georgia 30332

** Computer Science Department
Yale University
New Haven, Connecticut 06520

410 044

SPACE-TIME TRADEOFFS IN STRUCTURED PROGRAMMING: AN IMPROVED COMBINATORIAL EMBEDDING THEOREM

Richard A. DeMillo*
Stanley C. Eisenstat†
Richard J. Lipton†

* School of Information and Computer Science
Georgia Institute of Technology
Atlanta, GA 30332

† Computer Science Department
Yale University
New Haven, CT 06520

These results were announced at the 1976 Johns Hopkins Conference on Information Sciences and Systems. This research was supported in part by the U.S. Army Research Office, Grant Nos. DAHC04-74-G-0179 and DAAG29-76-G-0338; the Office of Naval Research, Grant No. N00014-67-097-0016; and the National Science Foundation, Grant No. DCR-74-12870.

Abstract: Let G and G* be programs represented by directed graphs. We define a relation $\leq_{S,T}$ between G and G* that formalizes the notion of G* simulating G with S-fold loss of space efficiency and T-fold loss of time efficiency, and prove that if G $\leq_{S,T}$ G*, where G has n statements and G* is structured, then in the worst case $T + \log_2 \log_2 S \geq \log_2 n + O(\log_2 \log_2 n)$.

Keywords and Phrases: ancestor tree, complexity, control structure, directed graph, embedding

CR Categories: 4.22, 4.34, 5.24, 5.32

1. Introduction

In a previous paper [1], we made precise some intuitive observations concerning the efficiency of structured programs by defining a combinatorial relation that corresponds to the notion of *uniform simulation* between programs. Informally, we say that a program G* uniformly simulates a program G if G* carries out the computation of G (and possibly additional computation which might be regarded as "bookkeeping") in such a way that the space-time efficiency of G is degraded by a factor that is independent of the size of G. The main results of [1] indicate that the non-existence of uniform simulations among many well-known classes of control structures is due to the combinatorial aspects of program structure and is not at all related to such details of program organization as choice of data structures or limitations on the form of Boolean expressions.

Indeed, the main result of [1] (Theorem 5.1) provides a non-trivial lower bound on the loss of space-time efficiency in *any* structured simulation of a goto program. This short note extends that result, improving the space-time inequality of [1, Theorem 5.1] by an exponential. Thus we now show that there are goto programs with n statements such that, for any structured simulation, either:

1) the simulation runs at least[†]

$$c_1 \log_2 n$$

times as slow as the original program,

or

2) the simulation has at least $2^{c_2 n^{c_3}}$ statements.

[†] We use c_1, c_2, c_3 to denote positive constants.

I.e., there are goto programs that can only be simulated by either *very slow* or *very large* structured programs.

In the sequel, we will concentrate on the combinatorial theorem that achieves these bounds. The programming language significance of the graphs and relations studied here is discussed extensively in [1].

2. Preliminaries

A *directed graph* G is an ordered pair (V,E) of *vertices* V and *edges* $E \subseteq V \times V$. A *path* in G is an ordered sequence of vertices connected by edges. For vertices $x,y \in V$, let $d_G(x,y)$ denote the length of a minimum length path form x to y. If no such path exists, then $d_G(x,y) = \infty$.

A *binary tree* is a directed graph that consists of either a single vertex or a *root* x and edges between x and the root of each of two binary trees called the left and right *subtrees* of x. A vertex x in a binary tree is a *leaf* if it has no sons. If H = (V,E) is a binary tree with root $r \in V$ and leaf $\ell \in V$, and $P = (x_1,\ldots,x_n)$ is a direct path from $x_1 = r$ to $x_n = \ell$, then P is called a *branch* of H. An *ancestor tree* G = (V,E) is a directed graph with the following properties:

1) There exists a subset $E_o \subseteq E$ such that $G_o = (V,E_o)$ is a binary tree;
2) If $(x,y) \in E - E_o$, then y is an ancestor of x in G_o.

Let G_n denote the $n \times n$ rook-connected array of vertices. If the vertices of G_n are indexed by (i,j) for $1 \le i,j \le n$, then, except for the obvious extremal conventions, there are symmetric edges between (i,j) and (i,j+1), (i+1,j).

For any directed graph G = (V,E), the notion of *boundary* makes sense. Let $A \subseteq V$. Then the *boundary of* A is defined as

$$\partial(A) = \{y \in V-A:\ \exists x \in A \text{ such that } (x,y) \in E\}$$

Clearly, $\partial(A)$ denotes the set of vertices not in A which are reachable from A by a single edge.†

By a simple improvement of a result from [1], we have the following important property of arrays:

Lemma 1: (Boundary Lemma) Let A be a set of vertices of G_n with $|A| \le n^2/2$. Then

$$2|A| \le |\partial(A)|^2.$$

3. Graph Embedding

The following relation was defined in [1]. Let $G = (V,E)$ and $G^* = (V^*,E^*)$ be directed graphs, and let $S,T > 0$. Then $G \le_{S,T} G^*$ if there is a partial function (called an *embedding*) $\phi: V^* \to V \cup \{\Lambda\}$, of the nodes of G^* to the nodes of G and a special node Λ, such that

1) $0 \le |\phi^{-1}(x)| \le S$ for all $x \in V$;
2) For all $x^* \in \phi^{-1}(V)$, if $d_{G^*}(\phi(x^*),y) < \infty$ for some $y \in V$, then there exists $y^* \in \phi^{-1}(y)$ such that $d_{G^*}(x^*,y^*) \le d_G(\phi(x^*),y)$.

If $\phi(v^*) = \Lambda$, then we refer to v^* as a *bookkeeping node*. If $\phi(v^*) = v \ne \Lambda$, then v^* is said to be a *copy* of v. Condition (1) states that there are at most S copies of any $v \in V$ in G^*. Condition (2) states that the embedding induces at most a T-fold increase in path length.

Theorem 1: [1, Theorem 5.2] If $S(n)$, $T(n)$ are such that $G_n \le_{S(n),T(n)} G^*$ for some ancestor tree G^*, then

$$T(n) + \log_2 S(n) \ge \log_2 n + c_1. \qquad (1)$$

The right hand side of inequality (1) cannot be improved, since with $S(n) \equiv 1$, the construction of [2] shows that

$$T(n) = O(\log_2 n)$$

† The notion of boundary used here corresponds to the coboundary of [1].

is achievable for any n vertex graph. Theorem 1, however, gives only a linear bound on S(n), and it has been conjectured that a non-polynomial lower bound on S(n) exists. In the next section we obtain such a bound.

4. Main Theorem

In this section, we obtain the following improvement of Theorem 1:

Theorem 2: If G^* is an ancestor tree and $G_n \leq_{S(n),T(n)} G^*$, then

$$T(n) + \log_2\log_2 S(n) \geq \log_2 n - O(\log_2\log_2 n).$$

Proof: For notational convenience, let us systematically confuse a graph with its set of vertices, so that "$x \in G$" and "$x \in V$" mean the same thing if $G = (V,E)$.

We assume $G_n \leq_{S,T} G^*$ via an embedding Φ. For any $A^* \subseteq G^*$, we use $\Phi(A^*)$ to denote the set of $x \in G_n$ which are Φ-images of some $x^* \in A^*$. Henceforth, we assume that G^* is a *binary tree*; it will be obvious as we progress that if G^* contains *ancestor edges*, then the proof is completely unaffected.

Let $P = (x_1^*, \ldots, {}_k^*)$ be a path of G^*. Then P is an *admissible path* if it is constructed as follows: For each x_i^* $(1 \leq i \leq k)$, let L_i^* denote the subtree of x_i^* containing x_{i+1}^*, and let R_i^* denote the other subtree of x_i^*; then either

a) $\Phi(R_i^*) \geq \Phi(L_i^*)$

or

b) $\Phi(R_i^*) \geq n^2/4$.

Note that the definition of admissible path is more general than that used in [1]. Indeed, it is by proving the existence of many such admissible paths that we obtain our result.

We fix an arbitrary admissible path $P = (x_1^*, \ldots, x_k^*)$ and define for $i = 1, \ldots, k$ the subtree $H_i^* = L_i^* \cup \{x_i^*\}$. We shall say that H_i^* is *small* if $|\Phi(H_i^*)| \leq n^2/4$; otherwise H_i^* is said to be *large*. Let

$$D_j = \bigcup_{\substack{1 \le i \le j \\ H_i \text{ is small}}} \phi(H_i^*);$$

in particular, D_k is the set of vertices in G_n which have copies in some small H_i^*.

Lemma 3: For some j,

$$\frac{n^2}{4} \le |D_j| \le \frac{n^2}{2}.$$

Proof: We need only show that there exists an integer j such that $|D_j| \ge n^2/4$, since if j is the least such integer, then (assuming $|D_o| = 0$)

$$|D_j| \le |D_{j-1}| + |\phi(H_j^*)| < \frac{n^2}{4} + \frac{n^2}{4} = n^2/2.$$

We claim that $|\phi(R_1^*)| \ge n^2/4$. For suppose otherwise, whence $|\phi(L_1^*)| \le |\phi(R_1^*)|$ by the definition of an admissible path. Now

$$\phi(G^*) = \phi(H_1^*) \cup \phi(R_1^*),$$

so that

$$n^2 = |\phi(G^*)| \le |\phi(L_1^*)| + 1 + |\phi(R_1^*)| \le 2|\phi(R_1^*)| + 1,$$

and thus

$$|\phi(R_1^*)| \ge n^2/4.$$

Let j be such that $|\phi(R_j^*)| = 0$, and let i be the largest integer such that $|\phi(R_i^*)| \ge n^2/4$. Then

$$|\phi(R_\ell^*)| < n^2/4, \text{ for } \ell = i+1,\dots,j.$$

Hence,

$$|\phi(H_\ell^*)| \le 1 + |\phi(L_\ell^*)| \le 1 + |\phi(R_\ell^*)| < 1 + n^2/4 \text{ for all } \ell = i+1,\dots,n.$$

But then each such H_ℓ^* is small, and therefore

$$\phi(R_i^*) \subseteq \bigcup_{i \le \ell \le j} \phi(H_\ell^*) \subseteq D_j.$$

But by the definition of i, $|D_j| \ge n^2/4$. $\square$

Letting k satisfy Lemma 3, we find that D_k satisfies the hypothesis of the Boundary Lemma, so that

$$|\partial(D_k)| \geq \sqrt{2}|D_k|^{1/2} \geq \frac{n}{\sqrt{2}}$$

Lemma 4: If ℓ_P is the number of large trees H_i^* along an admissible path P, then

$$\frac{n}{\sqrt{2}} \leq \ell_P\, 2^T.$$

Proof: Let

$$Q_T = \{v^* \in H_i^*, \text{ large: for some small } H_j^* \text{ and } x^* \in H_j^*,\ d_G(x^*,v^*) \leq T\}.$$

i.e., Q_T is the set of vertices in large H_i^* which are reachable from some node in a small H_j^* by a path of length at most T. We show that $|\partial(D_k)| \leq |Q_T|$ by defining an injection $g : \partial(D_k) \to Q_T$. For $y \in \partial(D_k)$, choose some $x \in D_k$ adjacent to y. Let x^* be a copy of x in a small H_j^*, let y^* be a copy of y such that $d_{G^*}(x^*,y^*) \leq T$, and set $g(y) = y^*$. Since $\Phi g(y) = \Phi(y^*) = y$, g is one-one. Thus, from (2),

$$|Q_T| \geq |\partial(D_k)| \geq \frac{n}{\sqrt{2}},$$

but

$$\begin{aligned} |Q_T| &\leq |\{H_i^* : H_i^* \text{ large}\}| \\ &\qquad \cdot |\{v^* : v^* \in H_i^*, \text{ large; } v^* \text{ within distance T of root of } H_i^*\}| \\ &\leq \ell_P \cdot 2^T \qquad \square \end{aligned}$$

To complete the proof, we now show that there are at least $2^{n/2^{T+1/2}}$ admissible paths. Since each admissible path corresponds to a distinct leaf† of G* and $G_n \leq_{S,T} G^*$, we have

$$2^{\frac{n}{\sqrt{2}}2^{-T}} \leq |\Phi^{-1}(V)| \leq S|V| = Sn^2$$

and the result follows.

---

† Without loss of generality, we assume that no leaf of G* is a bookkeeping node.

Lemma 5: There exist at least $2^{\ell_{min}}$ admissible paths, where $\ell_{min} = \frac{n}{\sqrt{2}} \cdot 2^{-T}$.

Proof: We prove the result by showing that at least ℓ_{min} independent binary choices must be made to construct an arbitrary admissible path. Consider a partial admissible path $x_1,\ldots,x_k$ (i.e., the initial segment of an admissible path). If only one subtree of x_k is large, then the admissible path can only be extended down that subtree. However, if both subtrees are large, then the admissible path can be extended down either subtree without violating the condition (a-b). By Lemma 4, there are at least ℓ_{min} large subtrees along every admissible path, and, for each such subtree, there is a node in the admissible path with two large subtrees. $\square\square$

By using the modeling strategy detailed in [1], we obtain the following:

Corollary: For each n there is an n statement goto program Q such that for *any structured simulation* of Q either

1) the simulating program is slower than Q by a factor of $c_1 \log n$, or

or

2) the simulating program is larger than Q by a factor of $2^{c_2 n^{c_3}}$.

An interesting interpretation of this result as a space-time tradeoff is shown in Figure 1, which illustrates, for fixed $n > 0$,

$$S(T,n) \geq 2^{n/2}$$

For any fixed value $K \leq T \leq c_1 \log n$, limiting the loss of time efficiency in the simulating program, the shaded region of Figure 1 shows the only values of S,T which are achievable.

Acknowledgements: We would like to thank Nancy Lynch, Ronald Rivest, Albert Meyer and Arnold Rosenberg for suggesting that we look for the improved embedding theorem contained in this paper.

References

1. R. J. Lipton, S. C. Eisenstat, and R. A. DeMillo, "Space-Time Hierarchies for Control Structures and Data Structures," *Journal of the ACM*, Vol. 23, No. 4, October 1976, pp. 720-737.

2. R. A. DeMillo, S. C. Eisenstat, and R. J. Lipton, "Preserving Average Proximity in Arrays," *Communications of the ACM*, Vol. 21, No. 3, March 1978, pp. 228-231.

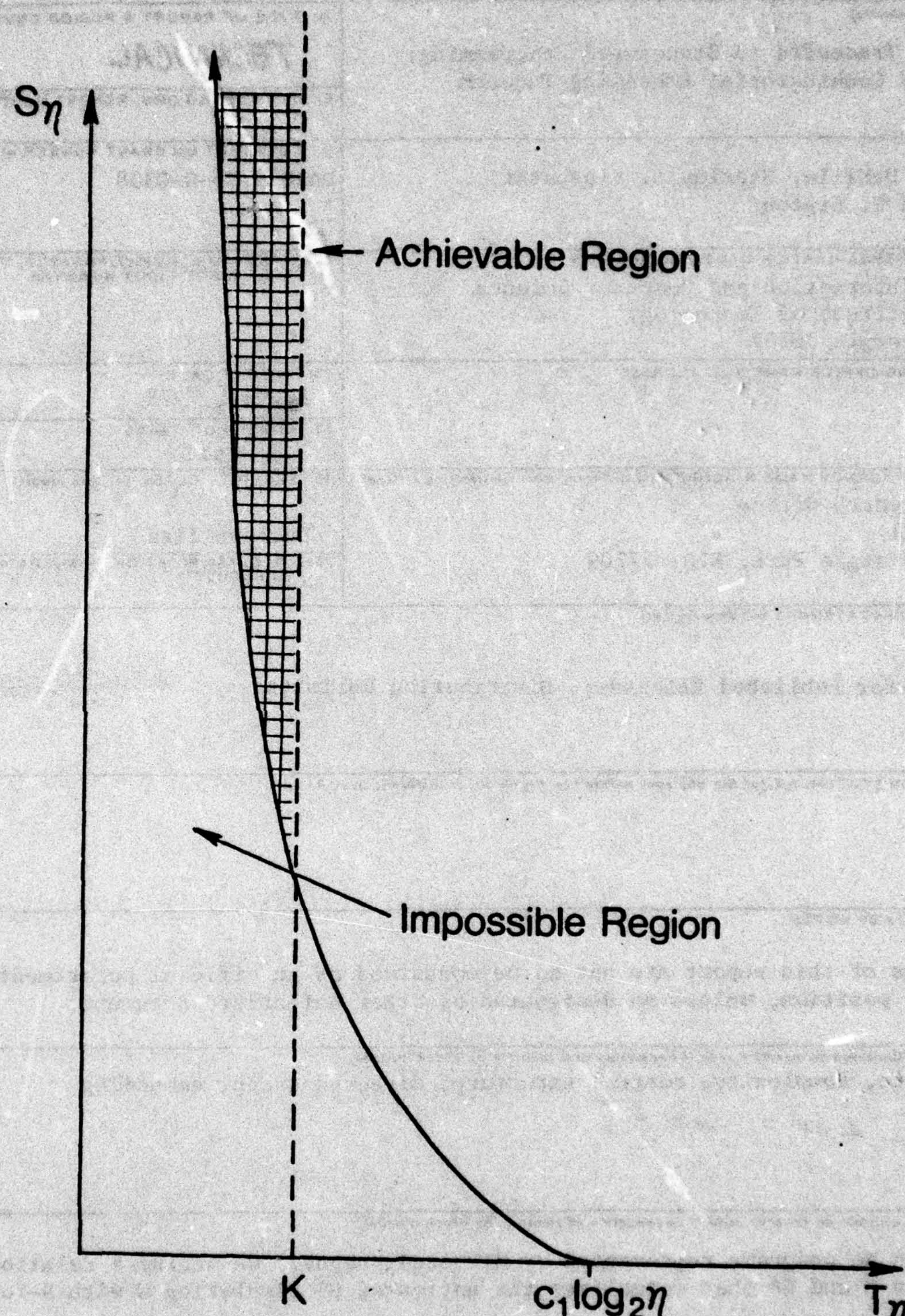

Figure 1. Trading-off T_η for $S_\eta = \left[2^{\eta/2.5}\right], 2^{-T_\eta}$ for fixed program size η.

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER ~~GIT-ICS-77/03~~	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) Space-Time Tradeoffs in Structured Programming: An Improved Combinatorial Embedding Theorem		5. TYPE OF REPORT & PERIOD COVERED TECHNICAL
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Richard A. DeMillo, Stanley G. Eisenstat, Richard J. Lipton		8. CONTRACT OR GRANT NUMBER(s) DAAG29-76-G-0338
9. PERFORMING ORGANIZATION NAME AND ADDRESS School of Information and Computer Science Georgia Institute of Technology Atlanta, Georgia 30332		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS		12. REPORT DATE August, 1978
		13. NUMBER OF PAGES 9 + iii
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) US Army Research Office PO Box 12211 Research Triangle Park, N.C. 27709		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for Published Releases; Distribution Unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

The findings of this report are not to be construed as an official Department of the Army position, unless so designated by other authorized document.

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

ancestor tree, complexity, control structure, directed graph, embedding

< or = sub S, T

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

Let G and G* be programs represented by directed graphs. We define a relation $\le_{S,T}$ between G and G* that formalizes the notion of G* simulating G with S-fold loss of space efficiency and T-fold loss of time efficiency, and prove that if G $\le_{S,T}$ G*, where G has n statements and G* is structured, then in the worst case $T + \log_2\log_2 S \geq \log_2 n + O(\log_2\log_2 n)$.

sub 2   sub 2   sub 2   sub 2

log sub 2(n)

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE